ESTELLE 명세에서 VHDL 명세로의 변환 방법론

이미경[†] · 이익섭[†] · 김선규[†] · 조준모[††] · 김성운[†††]

요 약

시스템 수준의 프로토콜 명세를 위한 정형 기법들은 S/W 구현에 중심을 둔 반면, 성능의 향상을 위해서 프로토콜의 설계를 H/W로 구현하는 것에 대한 필요성이 증대되고 있다. 따라서, 특정한 프로토콜을 이용한 시스템의 S/W적 구현에서 H/W적 구현으로 기반 환경이 변화함에 따라 IP(Integrated Processing)기술의 도입이 필요하게 되었다. 또한, 정형 명세로부터 시작하는 H/W의 구현 방법은 정형 명세의 특성에 의해, 구현된 H/W의 정확성, 신뢰성을 보장할 수 있다.

본 논문에서는 형식기술기법중의 하나인 ESTELLE로 정형화된 프로토콜을 H/W언어인 VHDL로 변환하는 방법을 제시한다. 우선ESTELLE 명세를 VHDL명세로 변환하는 변환 방법론을 기술한다. 이는 개념적인 변환방법으로 프로토콜 명세화의 기본 단위인 computation unit과 communication unit의 동작과 H/W에서의 프로세스간 동작의 유사성 등을 비교 분석하였다. 그 후 변환 모델을 기술하고, Inres 프로토콜을 통하여 실제로 ESTELLE을 VHDL로 변환하는 예를 제시한다.

Transformation Methodology from Specification of ESTELLE to VHDL

Mi Kyoung Lee[†], Ik Seob Lee[†], Seon Kyu Kim[†], Jun Mo Jo[††] and Sung Un Kim[†††]

ABSTRACT

Formal methods for protocol description of a system is based on the implementation of S/W. However, the importance of H/W implementation for a parts of protocol design is increasing. The combination between H/W and IP technology is needed since the implementation environment is changing from S/W to H/W for implementation of specific application protocol. H/W implementation method starting with formal description procedure is essential to guarantee correctness and reliability of the implemented H/W by characteristic of formal description language.

In this paper, for an automated H/W implementation, ESTELLE, a formal description method, is adopted. A transformation method from specification of ESTELLE to VHDL is suggested. This is an conceptual method that comparing and analyzing similarities between basic units of protocol description such as computation and communication unit and inter processors in H/W. Then we describe transformation model, and suggest example of transformation from ESTELLE to VHDL with Inres protocol.

1. 서 론

최근의 정보통신 관련 응용 프로토콜을 도입한 시스템들은 인터넷 보급이 확산됨에 따라 지능적인 서비스, 향상된 속도와 신뢰성을 제공하기 위해 더욱 복잡해지고 있다. 따라서 자연어로 기술된 프로토콜 명세는 자연어의 모호성, 모순성, 불완전성 등에 의해 신뢰성 있는 프로토콜 개발을 어렵게 하므로 정형기법의 사용이 필수적이며, 이는 검증, 구현 및 시험시에 전통적 자연어에 근거한 프로토콜 개발에 비해 더 체계적인 방법을 제공한다. 그림 1은 이러한 정형기법과 비정형기법을 이용하여 프로토콜을 개발하는 단계를 나타낸 것이다.

[†] 부경대학교 정보통신공학과 석사과정
[††] 동명대학 교수
[†††] 종신회원, 부경대학교 정보통신공학과 조교수

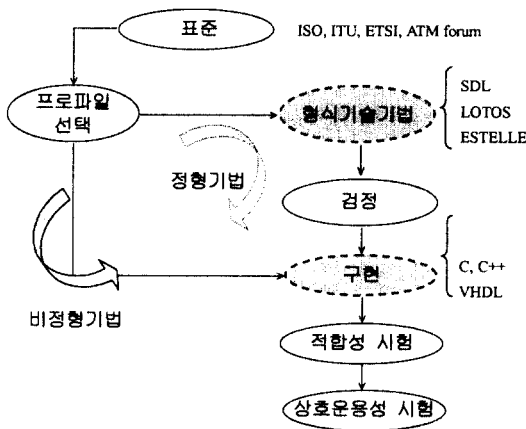



그림 1. 프로토콜 개발 과정

또한, 성능의 향상을 위해서 프로토콜의 설계를 H/W로 구현하는 것에 대한 필요성이 증대되고 있다. 오늘날 통신 프로토콜 설계의 어려움이 있으나 정형화된 명세로부터 H/W 구현을 실현 함으로써 프로토콜에 대한 기능적인 정확성과 신뢰성 뿐만 아니라 성능의 향상을 꾀할 수 있다. 현재 시스템 수준으로 명세된 통신 프로토콜을 H/W로 변환하는 설계 방법과 도구는 아직 존재하지 않으며 시스템 수준 명세 프로그램과 H/W설계를 위한 CAD 툴간의 연결을 위한 작업이 진행되고 있다[1].

본 논문은 통신 프로토콜의 정형 명세 언어인 ESTELLE로 정형화된 표현을 자동적으로 H/W 구현 언어인 VHDL 코드로 생성하는 방법론을 제시한다. 이를 위하여, 2장에서 ESTELLE과 VHDL개념을 기술하고, 3장에서는 ESTELLE에서 VHDL로 변환하는 방법론을 제시하고, 4장에서 두 언어 간의 의미론적 관계에 따라 ESTELLE의 VHDL로의 변환 모델을 소개한다. 5장에서는 실제 Inres프로토콜의 ESTELLE 명세를 VHDL로 변환한 예를 기술하고, 6장에서는 본 연구의 결론과 향후 연구 추진 사항에 대해 정리한다.

2. ESTELLE과 VHDL

2.1 ESTELLE

통신 시스템의 프로토콜 명세를 위한 정형명세 언어에는 ESTELLE, LOTOS, SDL등이 있다. ESTELLE은 ITU-T와 ISO가 공동으로 연구 개발한 형식적 명세 방법으로, FIFO 큐로 통신하는 확장된 유한상태천이 모델을 기본으로 한다. LOTOS는 Milner의 CCS(Calculus of Communicating System)를 확장한 개념으로 서비스와 프로토콜을 정의하기 위해 개발된 언어이다. 이는 간결하고 구조화된 기법으로 복잡한 분산 시스템을 높은 수준의 프로그래밍 언어로 기술된 모듈들의 집합으로 표현한다. SDL은 ITU-T에 의해 1976년에 표준화 되었으며 이는 초기에 전자교환기의 소프트웨어를 기술할 목적을 개발되었으나 통신망 프로토콜에도 적합하도록 확장되었다[2].

ESTELLE은 추상적인 데이터 타입 대신 Pascal과 비슷한 구조와 데이터 타입을 사용하기 때문에 SDL과 LOTOS 보다 덜 추상적이며, 이는 프로토콜의 제어부분 뿐만 아니라 데이터 타입부분을 다루는데 있어 다른 언어보다 융통성, 용이성 및 효율성이 높아 널리 이용되고 있다. 이에 본 논문에서는 정형명세 언어 중 ESTELLE을 VHDL로 변환하고자 한다[5][6].

ESTELLE에는 프로토콜의 기능을 표시하는 모듈이라는 개념과 모듈들간의 통신을 제공해주는 채널이라는 개념이 있다. 기본적인 구조적 요소는 모듈 인스턴스로 구성되며, 계층적 구조를 갖는 트리 형태로 구성된다. 이 모듈은 헤더와 본체로 구성되며, 헤더는 모듈의 외부 가시성을 정의하고 본체는 모듈의 자세한 행위를 정의한다. 이러한 모듈 헤더와 본체의 포함관계를 나타내는 ESTELLE 명세의 개요는 그림 2과 같다[3].

해당모듈과 자식모듈과의 동기를 결정하기 위해

Protocol Specification
Channel Definition
Module Type Definition
Module Body Definition
Types, Module's Variables, States, State Sets, Definitions
Initialized Part for the body
Transition Definition
Trans
From, to, when, provided,
Begin
{Action Pascal-like Statements}
end;
Module Variable Definition
Initialized Part for the Specification
Initialized Module Variables with Module Bodies
Channel Connection Definitions

그림 2. ESTELLE 명세의 개요

서 각 모듈은 반드시 systemprocess, systemactivity, process, activity 중 하나의 속성을 가져야 한다. System module은 독립적(비동기적)으로 동작한다. Process와 systemprocess 모듈의 자식모듈은 병렬로 동작하나, activity와 systemactivity 모듈의 자식모듈은 비결정적인 방법으로 동작한다. 즉, 하나의 자식 모듈의 천이가 종료되고, 다음에 다른 자식 모듈의 천이가 이루어진다.

ESTELLE에서는 모듈 간의 상호동작의 상호교환과 변수의 공유를 가능하게 하는 통신 메커니즘을 제공한다. 모듈은 각 상호동작점(IP: Inter-action Point)들 사이에 설정된 채널을 통하여 상호동작 한다. 또한 임의의 모듈과 그의 부모 모듈은 어떤 변수를 공유할 수 있는데 이것을 export변수라 한다. 공유변수는 부모와 자식이 동시에 액세스할 수 없으며, 부모/자식 우선순위 원칙에 의하여 부모가 우선순위를 갖는다.

모듈 인스턴스는 천이선택과 천이수행이라는 두 가지 동작을 형성하는 프로세스이다. 천이수행은 단일 computation으로 천이가 수행되는 동안에는 모듈 상태가 변하지 않고, 이 모듈 내에서 다른 동작이 수행되기 전에 수행중인 동작이 먼저 완료되어야 함을 의미한다. 천이 수행은 해당 천이 내에서 정의된 상태의 순차적인 computation이다. 선택단계에서는 모듈 내의 모든 천이 구절을 평가하여 만족시에 활성화(feasible)된 천이라고 부르는데, 모듈은 동작을 수행하기 위해 활성화된 천이들 가운데 하나를 선택한다.

2.2 VHDL

VHSIC (Very High Speed Integrated Circuits) Hardware Description Language (VHDL)은 1981년 미국방부 VHSIC Program Office로부터 파생되었다. 그 후 1987년에 VHDL Analysis and Standardization Group (VASG)에 의해 IEEE Standard-VHDL 1076으로, 약간의 정정과 추가가 이루어졌다. 이것은 개발, 검증, 합성, 하드웨어 설계의 테스팅, 하드웨어 설계 데이터의 통신 등을 지원한다. VHDL을 이용한 설계방법은 갈수록 복잡해지고 고집적화되는 H/W 설계 환경에서 보다 쉽게 회로를 설계할 수 있는 기회를 제공하여 현재 세계적으로 그 관심이 증가하고 있다[7].

VHDL의 기본적인 개념은 엔티티, 인스턴스와 시그널이다. 엔티티는 기능적인 블록, 출력된 회로보드, 집적회로, 마이크로 셀 또는 전(全) 시스템과 같은 하드웨어의 일부분으로 표현된다. VHDL 명세는 독립적으로 정의된 엔티티의 집합으로 구성되며, 엔티티 정의는 엔티티 선언과 아키텍쳐 선언으로 구성된다. 엔티티 선언은 외부적 가시성을 정의하고, 아키텍쳐 선언은 엔티티의 동작 또는 구조를 정의한다. 그림 3은 VHDL명세의 개요를 보여준다.

엔티티는 계층구조로 이루어져 있어 복잡한 하드웨어를 다루기 쉬운 모듈로 분해하며, 그것들 모두는 비동기적이며 병렬적으로 동작한다.

형제(sibling) 엔티티 사이, 또는 자식과 그 부모 엔티티 사이의 유일한 통신 수단은 시그널이다. 시그널은 주어진 타입 값의 혼합(과거의 값과 예정된 미래의 값)으로, 과거에 시그널이 가졌던 값 뿐만 아니라 현재 신호의 값에도 접근을 가능하게 한다. 또한 시그널은 공유된 변수로 인식되어 주어진 타입의 값 뿐만 아니라 그것의 실현 데이터 값으로도 저장된다. 외부 시그널은 엔티티 포트 리스트 선언 내에서 정의된다. 엔티티는 그것의 지역신호 또는 포트신호를 그것의 컴포넌트(자식 엔티티)의 포트 시그널로 연결시킨다. 그림 4는 시그널 연결과 그 명세의 일부분을 간단한 예로 보여준다.

보통 시그널은 하나의 드라이버를 다중 리시버로 연결시킨다. 시그널의 드라이버는 VHDL concurrent

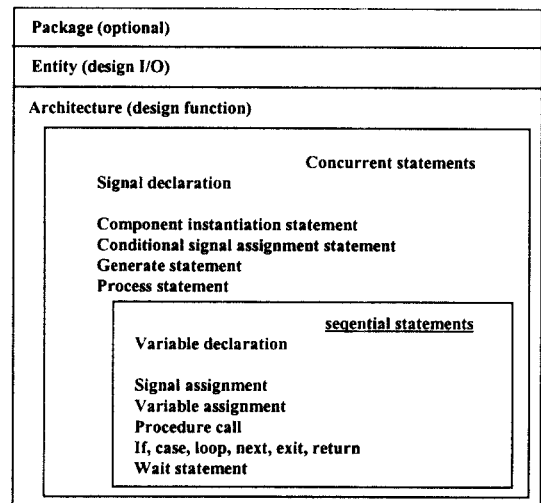



그림 3. VHDL명세의 개요

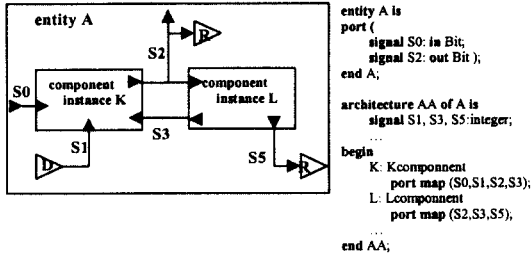



그림 4. VHDL 시그널 연결의 예

문으로 사례화(instantiated)되며, 이것은 왼쪽 편으로 신호를 할당하는 것이다. 리시버도 VHDL concurrent문으로 사례화 되는데, 리시버는 이러한 시그널을 테스트한다.

엔티티 동작은 관련된 아키텍쳐 선언내의 concurrent문에 의해 명세화 된다. 각 concurrent문은 action이고, 순차적인 프로그램에 의해 구현될 수 있다. 즉 다시 말해서 등가의 process문으로 쓰여질 수 있다. 구문론의 세부적인 사항을 모아보면, 엔티티 동작이 주어진 몇 개의 process문에 의해 정의됨을 알 수 있다. process는 그것의 조건이 만족되는 각 시간에 주기적으로 수행된다[8].

3. 변환 방법론

일반적으로 프로토콜 명세에서는 통신프로세스들 간의 협력을 정의 하는데, 그 프로세스들은 computation unit으로 동작하고, communication unit으로 통신 매체를 액세스하여 서로 통신한다. 그림5는 ESTELLE 요소와 computation unit과 communication unit사이의 관계를 나타낸다.

이 장에서는 통신 프로토콜의 H/W적 설계, 즉 computation unit과 communication unit의 동작과 H/W에서의 프로세스간 동작의 유사성을 논한다. 그리고, 이러한 유닛들이 모두 H/W에서의 수행되는 경우를 논하며 또한 communication unit이 H/W에서, 그리고 computation unit은 S/W에서 수행되는 경우도 논한다.

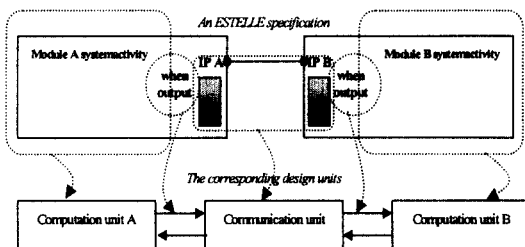



그림 5. ESTELLE 명세에서의 설계 개념도

3.1 H/W 접근법

H/W를 구현하는 요소는 어떤 기능적인 블록으로 호출된다. 기능적인 블록의 종류는 actor, 프로세서, 공유저장장치로 나누어진다. Actor는 하나의 프로세스로 구성되며 순차적인 작업을 형성, actor끼리 서로 통신하는 블록이다. 프로세서는 actor와는 달리 여러 개의 프로세스로 구성된 블록이다. 프로세스들 간의 통신은 공유저장 장치, 즉 레지스터와 메모리, 커패시터, 래치회로 등을 이용하여 이루어진다. 이 장치는 주어진 값을 유지하여, actor들 또는 프로세서들 사이에 넘겨주는 역할을 한다.

표 1과 그림 6은 ESTELLE개체와 H/W의 대응관계를 나타낸다.

Actor와 공유저장 장치의 인터페이스는 외부에 드러난 포트의 집합과 데이터 전송 메커니즘 모두를 가리킨다. 포트는 보통 이름, 데이터형, 방향등의 요소들로 속성화 된다. 통신 모델과 actor의 복잡도에 따라 데이터 전송에 다른 메커니즘이 적용되는데, 널

표 1. ESTELLE개체와 H/W의 대응관계

ESTELLE 개체	H/W
모듈	Actor 또는 프로세서
상호동작, signal	Data
큐	공유저장 장치
채널	전송link

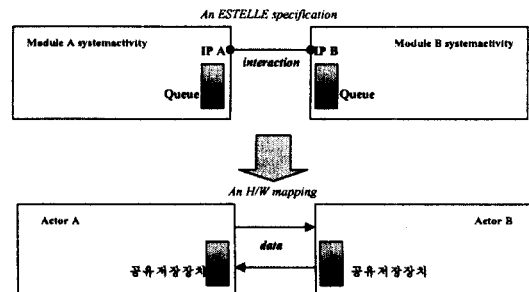



그림 6. ESTELLE개체와 H/W의 대응

리 사용되는 방법이 큐 메커니즘이다. 송신측 actor는 수신측 actor의 상태에 관계 없이 출력 데이터를 발생시킬 수 있는데, 큐는 이러한 actor들의 idle time을 최소화하기 위해 사용된다.

이와 같은 H/W의 내부동작을 볼 때, ESTELLE의 추상적인 통신 개체들은 여러 가지 방법으로 H/W에 대응된다. ESTELLE의 모듈은 서로간에 통신할 수 있는 프로세스로 동작한다는 점에서 H/W의 actor 또는 프로세서에 대응될 수 있으며, 상호동작은 시그널 또는 일반적인 데이터 패킷 형태이므로 H/W의 데이터 전송에 적용될 수 있다. 또한 ESTELLE의 큐는 데이터를 임시 저장하는 역할을 하므로 공유저장 장치로 구현된다. 실제의 큐는 ESTELLE 큐와는 달리 유한한 길이를 가지므로, 큐 오버플로우를 제어하기 위해 actor와 큐 사이의 동기화가 추가되어야 한다. 원하는 성능을 얻기 위해 적당한 큐 크기를 정할 수 있고, 수신된 상호동작의 크기가 커서 하나의 큐에 그 정보전체를 저장하기 어려울 수도 있으므로 큐에 그 상호동작의 위치를 나타내는 주소만 저장하는 방법을 사용한다.

ESTELLE 상호동작점 사이의 채널은 직렬 또는 병렬, 양방향 또는 단일방향의 데이터 전송링크로 구현될 수 있다. 기본적으로 ESTELLE 상호동작점 사이의 채널은 양방향으로 동작하고 분리가 가능하기 때문이다. H/W로의 구현 시 actor의 입력 포트, 출력 포트는 상호동작의 이름과 일치하고, 상호동작 이벤트와 상호동작의 인수값 등을 전송한다.

3.2 H/W 및 S/W 복합 접근법

두 개의 상호 동작하는 프로세스들 중 하나는 S/W에서 수행되고 다른 것은 H/W에서 수행될 때, S/W(device driver)와 H/W(H/W 블럭 인터페이스)간 통신 메커니즘이 필요하다. 전체 메커니즘에 있어 driver는 서브루틴이며 통신 프로그램과 함께 컴파일 된다. H/W 블록 인터페이스는 간단한 레지스터이거나 레지스터 집합일 수 있다. 이 레지스터는 마이크로프로세서 또는 컴퓨터 버스가 읽거나 쓴다.

H/W와 S/W사이에서 분산되는 ESTELLE모듈은 systemactivity나 systemprocess로 속성화 되어야 한다. 즉, ESTELLE 모듈들은 H/W나 S/W로 적절히 구현되고, 부가적인 동기화 작업을 피하기 위해 이와 같은 속성화가 필요하다. 이렇게 되면 상호동작만 H/W와 S/W 프로세스 사이의 통신에 사용된다. ESTELLE 상호동작은 데이터 패킷 형태로 구현된다.

H/W와 S/W 큐는 모두 똑같은 방법으로 구현된다. 즉, 둘 다 고정길이를 가지고, 전체 상호동작이나 주소들을 공동메모리에 저장할 수 있다. 큐는 H/W와 S/W에서 단일로 구현되거나 둘 사이에서 적당히 나뉠 수도 있다.

Device driver는 모듈을 구현한 H/W와 통신하기 위해 생성되고 모든 모듈의 상호동작점에 서비스를 제공한다. 상호동작점은 다른 모듈에 연결될 수 있고, 모듈은 같은 machine의 S/W에서 수행된다.

Device driver는 H/W 블럭 인터페이스와 직접 관계를 맺는다. H/W 블럭 인터페이스는 H/W 블록에서 나오거나 H/W 블록으로 들어가는 상호동작을 제공한다. 그림 7은 ESTELLE모듈을 S/W 및 H/W로 구성한 것이다.

위와 같은 여러 가지 논의를 종합해 볼 때, ESTELLE 개체들의 동작과 H/W 프로세스들의 동작의 유사성에 근거하여, ESTELLE 개체들은 H/W를 구현하는 언어인 VHDL로 변환이 가능하다.

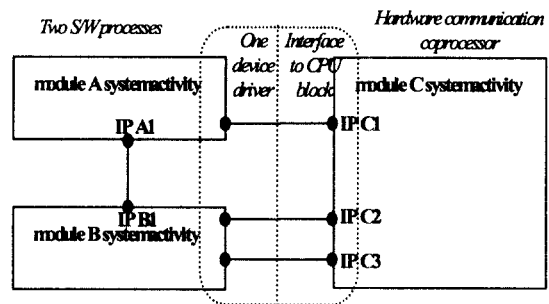



그림 7. S/W및 H/W모듈간 통신 개념도

4. ESTELLE의 VHDL변환 모델

이장에서는 ESTELLE과 VHDL 언어간의 의미론적 관계를 기술하고, ESTELLE로부터 VHDL로의 변환 모델을 소개한다. 변환모델은 두 부분으로 구성되며, 첫번째 부분은 ESTELLE 명세의 구조적인 요소, 즉 데이터 타입, 상수, 변수, 채널, 모듈, 상호동작점, 큐, export변수를 다룬다. 두번째 부분은ESTELLE 명세의 행위적인 요소, 즉 천이절, 선언부 등을 다룬다.

4.1 명세의 구조적인 부분

VHDL 엔티티는 ESTELLE 모듈에 대응되며, 컴포넌트 인스턴스는 자식 모듈 인스턴스를 표현한다. VHDL의 configuration 문은 컴포넌트 인스턴스와 엔티티를 서로 연결하며, ESTELLE 모듈의 외부적 관점은 VHDL 엔티티의 외부적 관점으로 변환되고 모듈의 내부적 관점은 엔티티의 내부적 관점으로 변환된다.

ESTELLE 채널 선언은 한쌍의 VHDL 타입으로 변환하며, 이 데이터 타입은 ESTELLE의 각 상호동작에 속하는 상호동작 이름과 그 파라미터를 가지고 있다. 이때 파라미터가 없다면 열거된(enumeration) 타입으로, 파라미터가 있으면 열거된 타입과 레코드 타입의 조합으로 기술된다. 그림 8은 ESTELLE 채널을 VHDL 신호 타입으로 변환한 예를 보여준다.

ESTELLE 모듈은 해더와 몸체 부분으로 구성되며, 해더는 VHDL의 엔티티 선언으로, 몸체는 아키텍처 선언으로 변환된다. ESTELLE init문은 모듈의 해더와 몸체 구조를 결정하여 모듈 인스턴스를 창조하며, 이것에 대응되는 엔티티 인스턴스는 VHDL compo-nent 사례화 문으로 기술된다. 이때 config-urationa문은 엔티티 인스턴스를 연결한다. 그림 9는 두 언어 구조의 매핑관계를 설명한다.

ESTELLE 모듈 해더는 상호동작점과 export 변수의 선언을 포함하고, 이 두 통신요소는 VHDL의 신호 포트 리스트 선언으로 변환된다. 모듈 해더는 모듈 인스턴스의 행위가 구별되도록 파리미터 리스트를 포함하고, 이는 같은 목적을 수행하는 VHDL generic 리스트로 변환된다. 그림 10은 ESTELLE 모듈내의 상호동작점이 VHDL의 신호로 변환되는 예를 보여준다.

ESTELL FIFO 큐는 모듈 인스턴스와 함께 생성되고, 외부내부 상호동작점의 선언으로 인해 그 큐가

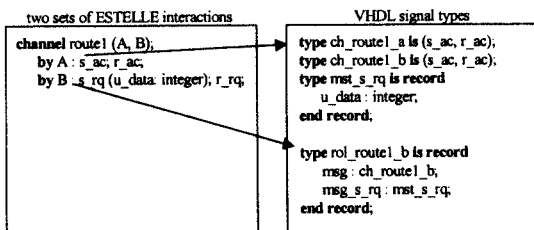



그림 8. ESTELLE 채널의 VHDL 신호타입 변환 예

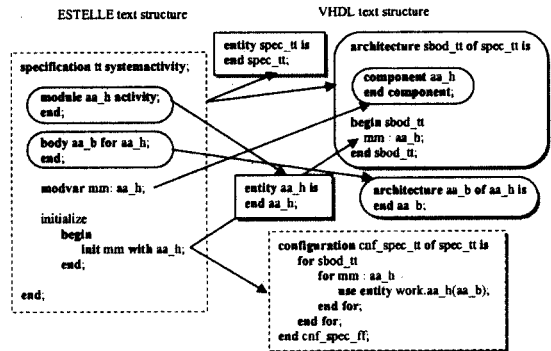



그림 9. 기술 구조의 변환

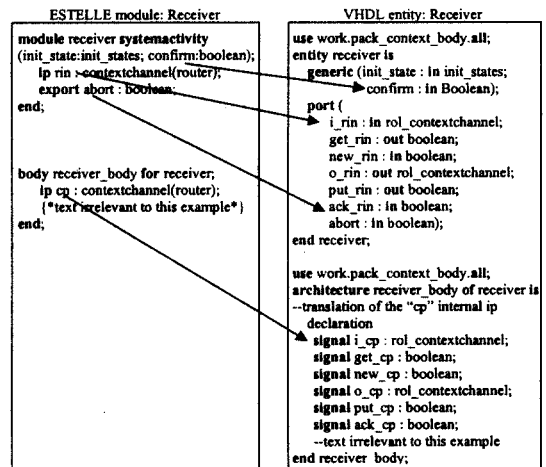



그림 10. 모듈 변환의 예

몇 개의 상호동작점에서 공동 또는 개별 큐로 사용되는지 식별가능하다. 이 큐는 VHDL에서 엔티티로 변환되는데, 그림 11에서 큐 엔티티의 구조를 보여준다.

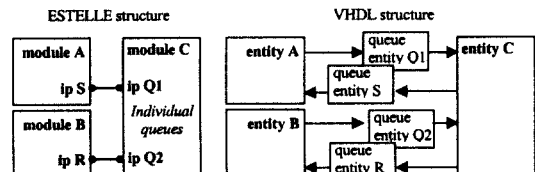



그림 11. ESTELLE 인터페이스의 VHDL 엔티티 연결 변환

4.2 명세의 행위적인 부분

ESTELLE 모듈은 순차적으로 처리되는데, 초기화 단계, 천이 선택과 천이 실행의 단계로 이루어진다. 모듈의 초기화 부분과 천이 부분은 모듈의 행위

를 기술하며 순차적으로 실행한다. 초기화 부분은 이것이 만들어진 인스턴스에서 한번씩 실행하며, 천이부분은 활성화된 천이가 비결정적 순서로 실행한다. 한 모듈 내에서는 천이의 병행이 없으므로 VHDL 프로세스 동작을 동시에 일어나도록 변환하지 않는다. 그림 12에서 보여지는 것처럼 이런 처리 과정을 VHDL에서는 하나의 프로세스 문으로 표현한다.

ESTELLE delay 절은 timeouts 을 다루기에 유용한 타이머를 나타낸다. VHDL로의 변환시 천이실행 시간은 상수 값을 가지며 이것은 computation과 관계되고, 또한 변수 값을 가질 수도 있으며 이것은 보내는 메시지의 크기와 수에 관계된다. 그림 13은 시간 파라미터의 변환 예를 보여준다.

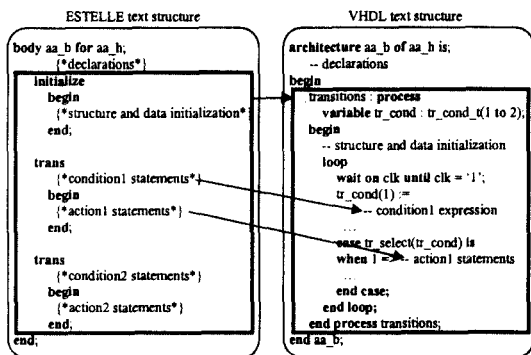



그림12. 행위적 기술 변환의 예

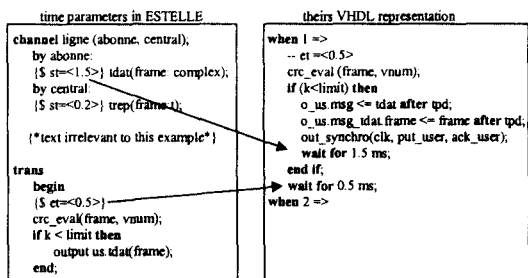



그림 13. 시간 파라미터의 응용 예

5. 변환의 예

5.1 Inres 프로토콜의 개요

Inres 프로토콜은 접속지향형 비동기 통신 프로토콜이다. Initiator는 연결을 설정하고 데이터를 전송하며, Responder는 연결을 해제하고 데이터를 수신한다. Inres는 크지 않아 이해하기 쉽고 connection, disconnection, sequence number, acknowledgement 및 시간초과(time-out)에 의한 재전송 등의 기본적인 OSI개념이 포함되어 있기 때문에 실제 시스템은 아니지만 시험목적으로는 적당한 프로토콜이다[4].

Inres 프로토콜 개체는 프로토콜 데이터 단위인 CR, CC, DT, AK와 DR를 교환함으로써 통신한다. 두개의 개체들간에 데이터 교환이 수행되기 위해서는 ICONreq, ICONind, ICONresp, ICONconf의 4개의 서비스 프리미티브를 교환함으로써 연결이 설정되어야 한다. 일단 연결이 성공적으로 설정되면 개시자는 IDATreq 서비스 프리미티브를 통해 데이터를 전송할 수 있고, IDATind 서비스 프리미티브에 의해 응답자에게 전달된다. 데이터 전송이 종료된 후에는 응답자에 의해 연결이 해제되는데 응답자가 요청한 IDISreq가 IDISind로 개시자에게 전달된다.

5.2 Inres 프로토콜의 ESTELLE명세화

Inres 프로토콜을 ESTELLE로 명세화하기 위해서는 먼저 모듈을 정의하여야 하고, 모듈간의 통신을 위해 채널과 상호동작점을 정의하여야 한다. 그림 15에서 나타낸 바와 같이 systemprocess의 속성을 갖는 5개의 모듈과 process의 속성을 갖는 4개의 모듈을 정의하고, ISAP 채널, Ipdu 채널, MSAP 채널을 설정하였으며 상호동작점인 ISAP, USER, PDU, MSAP, MSAP1, MSAP2를 표현하였다.

모듈이 형성된 채널의 상호동작점을 통하여 상호동작을 교환함으로써 모듈간에 원활한 통신이 이루어지므로 상호동작 리스트가 정의되어져야 한다. 또한 통신을 위해 사용되어지는 매개변수 리스트, 변수 등도 정의하여야 한다.

H/W 구성요소를 살펴보면 순차적인 작업을 수행하는 하나의 프로세스인 actor, 시분할 기술을 사용하여 병렬방식으로 동작하는 H/W 블록인 processor, 그리고 메모리, 레지스터, 래치회로 등의 값을 유지하고 정의된 인터페이스를 가지는 저장 장치로 구성되는데, 이는 ESTELLE에서 개별 큐, 천이부, 인터페이스(상호동작점과 채널)와 변수로 표현할 수 있다.

앞 절에서 기술된 방법을 바탕으로 하여 그림 15에 제시한 Inres 프로토콜 ESTELLE명세의 송신측

그림 15. ESTELLE 명세에서 Inres 프로토콜의 도식적 표현

station모듈을 VHDL블록 모듈로 변환하면 그림16과 같다.

그림 16. ESTELLE모듈의 VHDL로의 변환

이를 바탕으로 ESTELLE로 명세화 된 Inres 프로토콜을 모듈별로VHDL로 명세화 하면 다음과 같다.

- **Specification Structure**

Specification 부분은 VHDL에서 엔티티로 선언된다.

```
entity spec_inres is
end spec_inres;

architecture sbod_inres is
begin
end sbod_inres;

type ISDUType is ... ;
type Seqyencenumber is boolean;
type TduType is (CR, CC, DT, AK, DR);
type MSDUType is record
     Id : PduType;
     Num : Sequencenumber;
     Data : ISDUType;
end record;
```

- **VHDL Signal Types**

ESTELLE의 채널 부분은 VHDL에서 시그널 타입으로 선언된다.

```
type ISAPchn_USER is
   (ICONreq, ICONresp, IDATreq, IDISreq);
type ISDU is record
     ISDU : ISDUType;
end record;

type ISAPchn_USER_ is record
     msg : ISAPchn_USER;
     msg_IDATreq : ISDU;
end record;
```

- **Modules and Module Variables**

모듈의 선언과 변수의 정의를 보여준다.

```
entity initiator is
port ( i_USER : in ISAPchn;
       get_USER : out boolean;
       new_USER : in boolean;
       o_USER : out ISAPchn;
       put_USER : out boolean;
       ack_USER : in boolean;
       i_PDU : in Ipdu;
       get_PDU : out boolean;
       new_PDU : in boolean;
       o_PDU : out Ipdu;
       put_PDU : out boolean;
       ack_PDU : in boolean);
end initiator;

architecture initiator_Body of initiator is
type Cnt is range 0 to 4;
```

```
variable olddata : ISDUType;
variable counter : Cnt;
variable number : Sequencenumber;
function succ (Number : Sequencenumber)
          : Sequencenumber
    begin
      if Number = 0 then succ :=1
      else succ =0
    end;
state ini_state is (DISCONNECTED, WAIT,
    CONNECTED, SENDING);
signal state: ini_state

begin

end initiator_Body
```

6. 결 론

본 논문에서는 ESTELLE로 기술한 Inres 프로토콜을 H/W기술 언어인 VHDL로 변환하는 방법을 제시하였다. 프로토콜 개발의 정확성과 신뢰성을 보장하기 위한 정형기법과 더욱 빠른 속도를 위한 H/W 기술언어의 필요성에 바탕을 두고 있다. 또한, ESTELLE 모듈간의 통신을 일반적인 프로토콜 명세의 기본 단위인 computation unit과 communication unit간의 통신으로 유도하고, 이러한 기본 단위와 H/W 수준의 프로세스간 통신의 유사성을 비교 분석하였다. 그 결과를 바탕으로 명세된 프로토콜을 VHDL로 변환하였다.

H/W의 구현이 가능함에 따라 성능의 향상을 기대할 수 있으며 이로 인하여 일반적인 프로토콜 개발에 폭 넓게 쓰일 것으로 기대된다.

그러나, 형식기술언어와 H/W기술언어가 일대일 대응되는 것은 아니다. 즉, 각 언어가 가지고 있는 독특한 키워드가 하나의 키워드로 변환되지 않는다. 또한 근본적으로 S/W 바탕의 프로세스간 통신과 H/W 바탕의 프로세스간 통신은 같지 않기 때문에 두 언어간의 완전한 변환과 구현된 H/W의 정확성과 신뢰성을 보장하기 위한 연구가 필요하다.

참 고 문 헌

[1] Jacek Wytrebowicz, Contribution to the Methods of Protocol Hardware Implementation, 1995.

[2] ISO/IEC 9074, ESTELLE: A Formal Description Technique Based on an Extended State Transition Model, 1989.

[3] Lai R, An Experience in Using EDT to Process the ISO Transaction Processing ESTELLE Specification, Journal of Systems & Software, V.36 N.2, 1997.2.

[4] D. Hogrefe, OSI Formal Specification Case Study: the Inres Protocol and Service, Revised, Technical Report, University of Berne, Institute for Informatics, pp 33, May 1992.

[5] 한국전자통신연구원, 정보통신 프로토콜 공학, 1994. 8.

[6] 한국전자통신연구원, 프로토콜 공학과 프로토콜 시험, 1997.8.

[7] 홍릉과학출판사, VHDL 기초와 응용, 1997.4.

[8] Fabrizion Ferrandi, Franco Fummi, Donatella Sciuto, Implicite Test Generation for Behavioral VHDL Models Proceedings of the International Test Conference 1998, p.587-596, 1998. 10

이 미 경

2000년 2월 부경대학교 정보통신공학과 졸업(공학사)
2000년 3월 부경대학교 정보통신공학과 석사과정 재학중
관심분야 : 프로토콜 공학, Optical Network

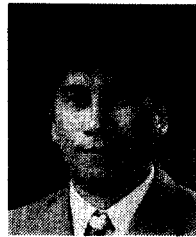
이 익 섭

2000년 2월 부경대학교 정보통신공학과 졸업(공학사)
2000년 3월 부경대학교 정보통신공학과 석사과정 재학중
관심분야 : 프로토콜 공학, IMT-2000, 무선망

김 선 규

2000년 2월 부경대학교 정보통신공학과 졸업(공학사)
2000년 3월 부경대학교 정보통신공학과 석사과정 재학중
관심분야 : 프로토콜 공학, 한글 필기체 인식

조 준 모

95년과 97년에 경북대학교에서 석사학위 및 박사과정을 수료한 후, 98년이후 동명대학에 교수로 근무하고 있음. 관심분야는 ATM 교환기, VHDL임.

김 성 운

90년과 93년에 프랑스 파리 7대학 석사 및 박사학위를 취득하였다. 82년부터 85년까지 한국전자통신연구원에서 근무하였으며, 86년부터 95년까지 한국통신연구개발원에서 근무하였음. 96년 이후 부경대학교 정보통신공학과 조교수로 재직중임.
관심분야 : 초고속망 및 프로토콜공학분야임.